ធនាគារជាតិ នៃ កម្ពុជា
ប្រាក់រៀល. ស្ថិរភាព. អភិវឌ្ឍន៍.

Implementation Guideline

KHQR SDK Documentation

Version 2.7

DOCUMENT HISTORY

Version	Date	Description
1.0	14.12.2020	
1.0.1	04.01.2021	- Alignment content. - Remove the Error Handling part. - Make all contents in Requirement, Installation and Usage parts more consistent.
1.1	25.01.2021	- Add a timestamp to the QR code string. - Add md5 hash string to response data.
1.1.1	01.02.2021	- Add Bill Number, Store Label, Terminal Label in QR - Add the request parameter section.
1.2	17.02.2021	Adding decode KHQR information function.
1.3	15.03.2021	Add Javascript SDK
1.4	20.04.2021	- Add function to generate KHQR deeplink. - Rename decode function name from **decodeQR** to **decode**. (iOS only)
1.5	08.07.2021	Add deeplink routing diagram
1.6	11.07.2021	- Add merchant Id for merchant type. - Add Acquiring Bank merchant name for merchant type. - Add a mobile number to additional data. - Capture value of merchant city. - Update generate KHQR function. - Update decode response value. - Add deeplink routing diagram.
1.7	01.09.2021	- Update generate individual by adding new tag for account information and acquiring bank name. - Update decode function without validate on required fields.
1.8	12.11.2021	- Make improvements on the verify function for iOS, Java and javascript.
1.9	23.11.2021	- Fix an issue on generating KHQR of an amount with leading zero.
2.0	04.01.2021	Moving library for stable API request on javascript SDK and improve verify KHQR content function for all SDK.
2.1	13.01.2022	Fix bug on decode function in javascript SDK
2.2	28.03.2022	Fix bug on decode function in javascript SDK
2.3	20.04.2022	Fix an issue on the verify function from get invalid status to valid status when QR contains any optional subtags that are not defined in standard KHQR document.
2.4	05.09.2022	- Add changelogs on iOS SDK v 1.0.0.11 - Add a new error code.
2.5	20.02.2023	- Change package name from .org to .gov - Update phone number validation
2.6	19.05.2023	Remove example KHQR content that using main wallet

2.7	05.07.2023	- Add UPI tag - Add Purpose of Transaction tag - Add Merchant Alternate Language tag

TABLE CONTENTS

Introduction

Overview

The standardization of KHQR code specification will help promote wider use of mobile retail payments in Cambodia and provide consistent user experience for merchants and consumers. It can enable interoperability in the payment industry. A common QR code would facilitate payments among different schemes, e-wallets and banks and would encourage small merchants to adopt KHQR code as payment method.

KHQR is created for retail or remittance in Cambodia and Cross-Border. It only requires a single QR for receiving transactions from any payment provider through Bakong including Bakong App. For more detail please refer to **Prakas KHQR Code Specification** in Cambodia.

Purpose

This document describes the detailed specification of how to use KHQR SDK offered by **National Bank of Cambodia**. The expected readers are NBC technical team and third-party technical team. This can be used as reference for any interest related to the KHQR SDK.

Scope

This document contains the complete description of the KHQR SDK specification including: Features, Requirement, Installation Guide, Usage and FAQ.

Version Control

SDK	Version	Description
Javascript	1.0.0	Having generate decode and verify KHQR feature
	1.0.1	Add generate deeplink feature
	1.0.2	- Ghost Version
	1.0.3	- Add merchant Id for merchant type. - Add Acquiring Bank merchant name for merchant type. - Add a mobile number to additional data. - Capture value of merchant city. - Update generate KHQR function. - Update decode response value.
	1.0.4	- Ghost version
	1.0.5	- Update generate individual by adding new tag for account information and acquiring bank name. - Update decode function without validate on required fields - Fix amount formatting
	1.0.6	- Adding a function to check bakong account
	1.0.7	Make improvements on the verify function.
	1.0.8	Moving library for stable API request
	1.0.9	Improve verify KHQR content function.
	1.0.10	Fixed on decode function (Return null data)
	1.0.11	Fixed on decode cache issue
	1.0.12	- Ghost version
	1.0.13	- Fixed on wrong decode information and issue with timestamp without subtag on decode function. - Fix null value on optional data does not throw exceptions anymore.
	1.0.14	Update phone number validation.
	1.0.15 **(Latest)**	- Added UnionPay Tags - Added merchant information-language template - Added purpose of transaction in AdditionalData Information
iOS	0.1.1	

	1.0.0.1	- Add a timestamp to the QR code string. - Add md5 hash string to response data.
	1.0.0.2	- Add Bill number, Store label and terminal label to QR String.
	1.0.0.3	Add decode KHQR information function.
	1.0.0.4	Changes on function decode khqr - Validate length and CRC - Validate each tag according to KHQR specification
	1.0.0.5	Add generate deep link function
	1.0.0.6	- Add merchant Id for merchant type. - Add Acquiring Bank merchant name for merchant type. - Add a mobile number to additional data.. - Capture value of merchant city. - Update generate KHQR function. - Update decode response value.
	1.0.0.7	- Update generate individual by adding new tag for account information and acquiring bank name. - Update decode function without validate on required fields
	1.0.0.8	Make improvements on the verify function.
	1.0.0.9	Improve verify KHQR content function.
	1.0.0.10	Fix an issue on the verify function from get invalid status to valid status when QR contains any optional subtags that are not defined in standard KHQR document.
	1.0.0.11	1. Fixed wrong amount with precision 2. Validate tag length. 3. Validate duplicate tag.
	1.0.0.12	Change package name from .org to .gov
	1.0.0.13	Update phone number validation
	1.0.0.14 **(Latest)**	- Added UnionPay Tags - Added merchant information-language template - Added purpose of transaction in AdditionalData Information
Java	1.0.0.4	Add decode function
	1.0.0.5	Add generate deep link function

	1.0.0.6	- Add merchant Id for merchant type. - Add Acquiring Bank merchant name for merchant type. - Add a mobile number to additional data.. - Capture value of merchant city. - Update generate KHQR function. - Update decode response value.
	1.0.0.7	- Update generate individual by adding new tag for account information and acquiring bank name. - Update decode function without validate on required fields
	1.0.0.8	Make improvements on the verify function.
	1.0.0.9	Improve verify KHQR content function.
	1.0.0.10	- Change package name from .org to .gov - Update phone number validation
	1.0.0.11 **(Latest)**	- Added UnionPay Tags - Added merchant information-language template - Added purpose of transaction in AdditionalData Information
C#	0.1.1	
	1.0.0.1	- Add a timestamp to the QR code string. - Add md5 hash string to response data.
	1.0.0.2	- Add Bill number, Store label and terminal label to QR String.
	1.0.0.3	Add decode KHQR information function.
	1.0.0.4	Changes on function decode khqr - Validate length and CRC - Validate each tag according to KHQR specification
	1.0.0.5	Add generate deep link function
	1.0.0.6	- Add merchant Id for merchant type. - Add Acquiring Bank merchant name for merchant type. - Add a mobile number to additional data.. - Capture value of merchant city. - Update generate KHQR function. - Update decode response value.
	1.0.0.7	- Update generate individual by adding new tag for account information and acquiring bank name.

		- Update decode function without validate on required fields
1.0.0.8		Fix an issue on generating KHQR of an amount with leading zero.
1.0.0.9		Improve verify KHQR content function.
1.0.0.10		Change package and namespace from org to gov
1.0.0.11		Update phone number validation.
1.0.0.12 (Latest)		- Added UnionPay Tags - Added merchant information-language template - Added purpose of transaction in AdditionalData Information

Features

- Generate KHQR.
- Verification (Valid or Invalid).
- Decode KHQR Information.
- Generate KHQR Deeplink.

Requirement

iOS

- Development target 11.0 +
- Dependency management using Cocoapod

Java

- Source compatibility Java 8
- Target compatibility Java 8

C#

- NetStandard 2.0
- .Net Framework 4.0 and later
- .NET Core 2.0 and later
- Xamarin.iOS
- Xamarin.Android

Javascript

- NPM Javascript package management
- script raw file

Installation

iOS

CocoaPods

- **How to**

    1. Add source to podfile

    ```
    source "https://sambo:ycfXmxxRbyzEmozY9z6n@gitlab.nbc.gov.kh/khqr/khqr-ios-pod.git"
    ```

    2. Add pod name to podfile

    ```
    pod "BakongKHQR"
    ```

    3. Run command

    ```
    pod install
    ```

Java

Maven

- **How to**

    1. Add dependency to maven **pom.xml**

    ```xml
    <dependency>
        <groupId>kh.gov.nbc.bakong_khqr</groupId>
        <artifactId>sdk-java</artifactId>
        <version>current_version</version>
    </dependency>
    ```

Gradle

- **How to**

    1. Add dependency to **build.gradle**

    ```
    dependencies {
            implementation 'kh.gov.nbc.bakong_khqr:sdk-java:{current_version}'
    }
    ```

C#

Nuget

- **How to**

    1. Using Visual Studio Package Manager

    ```
    Install-Package Kh.Gov.Nbc.BakongKHQR
    ```

    2. Using Dotnet Cli

    ```
    dotnet add package Kh.Gov.Nbc.BakongKHQR
    ```

    3. Include in csproj

    ```
    <ItemGroup>
      <PackageReference Include="Kh.Gov.Nbc.BakongKHQR" Version=current_version
    />
      <!-- ... other packages goes here -->
    </ItemGroup>
    ```

Install Manually

If you prefer not to use nuget, you can integrate this SDK into your project manually via dll file.

- **How to**

    1. Go to Nuget page of Kh.Gov.Nbc.BakongKHQR

       https://www.nuget.org/packages/Kh.Gov.Nbc.BakongKHQR

2. Download package (nupkg file).

3. Extract the package using any unzip tool.

4. Copy the dll file based on your platform from the following folder.

   For .NetFramework 4.0 copy dll from this path

   ```
   lib/net40/kh.gov.nbc.bakong_khqr.dll
   ```

   For .NetFramework 4.5 copy dll from this path

   ```
   lib/net45/kh.gov.nbc.bakong_khqr.dll
   ```

   For .NetFramework 4.6 copy dll from this path

   ```
   lib/net46/kh.gov.nbc.bakong_khqr.dll
   ```

   Other Platforms (.NetStandard 2.0) copy from this path

   ```
   lib/netstandard2.0/kh.gov.nbc.bakong_khqr.dll
   ```

   Please consult the NetStandard support table for further explanation.
   https://docs.microsoft.com/en-us/dotnet/standard/net-standard#net-implementation-support

Javascript

NMP

- **How to**

  1. Init project with NPM

     ```
     npm init -y
     ```

2. Install KHQR SDK

```
npm install bakong-khqr
```

3. Use package

```
const {BakongKHQR, khqrData, IndividualInfo, MerchantInfo, SourceInfo} =
require("bakong-khqr");
```

Or

```
import {BakongKHQR, khqrData, IndividualInfo, MerchantInfo, SourceInfo} from
"bakong-khqr";
```

Raw Script

- **How to**

  1. Link script to HTML

  ```
  <script
  src="https://drive.google.com/uc?export=view&id=12emdSpcYBx4NyOm-siW-OnD
  y0HEpdR-4">\
  </script>
  ```

  2. Use it at .js file

  ```
  let KHQR = BakongKHQR
  // This is the object consist of BakongKHQR, khqrData, IndividualInfo, MerchantInfo,
  SourceInfo
  ```

Usage

Request Parameter

- **Generate KHQR (Individual)**

Parameter	Type	Mandatory	Length	Description
BakongAccountID	String	Y	32	
AccountInformation	String	N	32	Account number or phone number.
AcquiringBank	String	N	32	Acquiring Bank name.
Currency	KHQRCurrency	N		Default: KHR
Amount	Double	N	13	
MerchantName	String	Y	25	
MerchantCity	String	N	15	Default: Phnom Penh
BillNumber	String	N	25	
MobileNumber	String	N	25	example: 855967854321
StoreLabel	String	N	25	
TerminalLabel	String	N	25	
purposeOfTransaction	String	N	25	
upiAccountInformation	String	N	31	example: 0001034400010344 ABCDEFGHIJKLMNO which contain: Acquirer IIN: 00010344 Forwarding IIN: 00010344 Merchant ID: ABCDEFGHIJKLMNO
merchantAlternateLanguagePreference	String	N	2	
merchantNameAlternateLanguage	String	N	25	
merchantCityAlternateLanguage	String	N	15	

- **Generate KHQR (Merchant)**

Parameter	Type	Mandatory	Length	Description
BakongAccountID	String	Y	32	
MerchantID	String	Y	32	
AcquiringBank	String	Y	32	Acquiring Bank name.
Currency	KHQRCurrency	N		Default: KHR
Amount	Double	N	13	
MerchantName	String	Y	25	
MerchantCity	String	N	15	Default: Phnom Penh
BillNumber	String	N	25	
MobileNumber	String	N	25	example: 855967854321
StoreLabel	String	N	25	
TerminalLabel	String	N	25	
purposeOfTransaction	String	N	25	
upiAccountInformation	String	N	31	example: 0001034400010344 ABCDEFGHIJKLMNO which contain: Acquirer IIN: 00010344 Forwarding IIN: 00010344 Merchant ID: ABCDEFGHIJKLMNO
merchantAlternateLanguagePreference	String	N	2	
merchantNameAlternateLanguage	String	N	25	

- **Verification KHQR**

Parameter	Type	Mandatory	Length	Description
qrCode	String	Y		

- **Decode KHQR**

Parameter	Type	Mandatory	Length	Description
qrCode	String	Y		

- **KHQR Deeplink**

Parameter	Type	Mandatory	Length	Description
qr	String	Y		
url	String	Y		URL for API generate Deeplink. Provided by NBC team.
sourceInfo		N		
appIconUrl	String	Y		Your app icon
appName	String	Y		Your app name
appDeepLinkCallBack	String	Y		Deeplink url for opening your app after payment is completed.

iOS

- **Generate KHQR for Individual**

Swift

```swift
let info = IndividualInfo(accountId: "john_smith@devb",
                          merchantName: "John Smith",
                          accountInformation: "85512233455",
                          acquiringBank: "Dev Bank",
                          currency: .Usd,
                          amount: 100)
info?.billNumber = "#12345"
info?.mobileNumber = "85512233455"
info?.storeLabel = "Coffee Shop"
info?.terminalLabel = "Cashier_1"
info?.purposeOfTransaction = "Morning coffee"
```

```swift
info?.upiAccountInformation = "12345678123456789012345"

info?.merchantAlternateLanguagePreference = "km"

info?.merchantNameAlternateLanguage = "ចន ស្ម៊ីន"

info?.merchantCityAlternateLanguage = "សៀមរាប"

let khqrResponse = BakongKHQR.generateIndividual(info!)

if khqrResponse.status?.code == 0 {

    let khqrData = khqrResponse.data as? KHQRData

    print("data: \(khqrData?.qr)")

    print("md5: \(khqrData?.md5)")

}
```

Objective C

```objc
IndividualInfo * info = [[IndividualInfo alloc] initWithAccountId: @"john_smith@devb"

                                        merchantName: @"John Smith"

                                        accountInformation: @"85512233455"

                                        acquiringBank: @"Dev Bank"

                                        currency: Usd

                                        amount: 100];

[info setBillNumber: @"#12345"];

[info setMobileNumber: @"85512233455"];

[info setStoreLabel: @"Coffee Shop"];

[info setTerminalLabel: @"Cashier_1"];

[info setPurposeOfTransaction: @"Morning coffee"];

[info setUpiAccountInformation: @"12345678123456789012345"];

[info setMerchantAlternateLanguagePreference: @"km"];

[info setMerchantNameAlternateLanguage: @"ចន ស្ម៊ីន"];

[info setMerchantCityAlternateLanguage: @"សៀមរាប"];

KHQRResponse * khqrResponse = [BakongKHQR generateIndividual: info];

if (khqrResponse.status.code == 0) {

    KHQRData * khqrData = (KHQRData *) khqrResponse.data;

    NSLog(@"data: %@", khqrData.qr);

    NSLog(@"md5: %@", khqrData.md5);

}
```

Result

data:
00020101021215231234567812345678901234529460015john_smith@devb0111855122334550208Dev bank5204599953038405406100.005802KH5910John Smith6010Phnom Penh62710106#1234502118551223345550311Coffee Shop0709Cashier_10814Morning coffee64280002km0108ចិន ស្មីន0206សៀមរាប991700131688006997619630042B5B

md5: 3f3b8ba2ea4d044accb950f797a65a12

- **Generate KHQR for Merchant**

Swift

```swift
let merchantInfo = MerchantInfo(accountId: "khqr@devb",
                                merchantId: "123456",
                                merchantName: "John Smith",
                                acquiringBank: "Dev Bank",
                                currency: .Usd,
                                amount: 100)
merchantInfo?.billNumber = "#12345"

merchantInfo?.mobileNumber = "85512233455"

merchantInfo?.storeLabel = "Coffee Shop"

merchantInfo?.terminalLabel = "Cashier_1"

merchantInfo?.purposeOfTransaction = "Morning coffee"

merchantInfo?.upiAccountInformation = "12345678123456789012345"

merchantInfo?.merchantAlternateLanguagePreference = "km"

merchantInfo?.merchantNameAlternateLanguage = "ចិន ស្មីន"

merchantInfo?.merchantCityAlternateLanguage = "សៀមរាប"

let khqrResponse = BakongKHQR.generateMerchant(merchantInfo!)

if khqrResponse.status?.errorCode == 0 {

    let khqrData = khqrResponse.data as? KHQRData

    print(khqrData?.qr)

    print(khqrData?.md5)

}
```

Objective C

```objectivec
MerchantInfo * merchantInfo = [[MerchantInfo alloc] initWithAccountId: @"khqr@devb"
 merchantId: @"123456"
merchantName: @"John Smith"
acquiringBank: @"Dev Bank"
currency: Usd
amount: 100];
[merchantInfo setBillNumber: @"#12345"];
[merchantInfo setMobileNumber: @"85512233455"];
[merchantInfo setStoreLabel: @"Coffee Shop"];
[merchantInfo setTerminalLabel: @"Cashier_1"];
[merchantInfo setPurposeOfTransactionl: @"Morning coffee"];
[merchantInfo setUpiAccountInformation: @"12345678123456789012345"];
[merchantInfo setMerchantAlternateLanguagePreferencel: @"km"];
[merchantInfo setMerchantNameAlternateLanguage @"ចន ស្មីន"];
[merchantInfo setMerchantCityAlternateLanguage: @"សៀមរាប"];
KHQRResponse * khqrResponse = [BakongKHQR generateMerchant: merchantInfo];
if (khqrResponse.status.code == 0) {
    KHQRData * khqrData2 = (KHQRData *) khqrResponse.data;
    NSLog(@"data: %@", khqrData2.qr);
    NSLog(@"md5: %@", khqrData2.md5);
}
```

Result

```
data:
00020101021215231234567812345678901234530350009khqr@devb010612345602
08Dev Bank520459995303840540610.005802KH5910John Smith6010Phnom
Penh62710106#12345021185512233455503111Coffee Shop0709Cashier_10814Morning
coffee64280002km0108ចន ស្មីន0206សៀមរាប991700131688007660712063049740

md5: debe4bb15966de7d30c878f9dda5f3a7
```

- **Verification KHQR**

Swift

```swift
let qrCode =
"00020101021215231234567812345678901234529460015john_smith@devb0111855
122334550208Dev bank520459995303840540610 0.005802KH5910John
Smith6010Phnom Penh62710106#123450211855122334550311Coffee
Shop0709Cashier_10814Morning coffee64280002km0108ចិន
ស្ពៀន0206សៀមរាប99170013168800699761963042B5B"

let response = khqr?.verify(qrCode)

let crcValidation = response?.data as? CRCValidation

print("valid : \(crcValidation?.valid)")
```

Objective C

```objc
NSString *qrCode =
@"00020101021215231234567812345678901234529460015john_smith@devb01118
55122334550208Dev bank520459995303840540610 0.005802KH5910John
Smith6010Phnom Penh62710106#123450211855122334550311Coffee
Shop0709Cashier_10814Morning coffee64280002km0108ចិន
ស្ពៀន0206សៀមរាប99170013168800699761963042B5B";

KHQRResponse * response = [khqr verify: qrCode];

CRCValidation* crcValidation = (CRCValidation *) response.data;

NSLog(@"valid: %d", crcValidation.valid);
```

Result

```
valid: 1
```

- **Decode KHQR**

Swift

```swift
let decodeResponse =

BakongKHQR.decode("00020101021215231234567812345678901234529460015john_
smith@devb0111855122334550208Dev
bank5204599953038405406100.005802KH5910John Smith6010Phnom
Penh62710106#123450211855122334550311Coffee Shop0709Cashier_10814Morning
coffee64280002km0108ចិន ស្ម៊ីន0206សៀមរាប99170013168800699761963042B5B")

let decodeData = decodeResponse?.data as? KHQRDecodeData

decodeData?.printAll()
```

Objective C

```objc
KHQRResponse* decodeResponse = [BakongKHQR decode:
@"00020101021215231234567812345678901234529460015john_smith@devb01118
55122334550208Dev bank5204599953038405406100.005802KH5910John
Smith6010Phnom Penh62710106#123450211855122334550311Coffee
Shop0709Cashier_10814Morning coffee64280002km0108ចិន
ស្ម៊ីន0206សៀមរាប99170013168800699761963042B5B"];

KHQRDecodeData* decodeData = (KHQRDecodeData *) decodeResponse.data;

[decodeData printAll];
```

Result

```
payloadFormatIndicator = 01

pointOfInitiationMethod = 12

merchantType = 29

upiAccountInformation = 12345678123456789012345

bakongAccountID = john_smith@devb

merchantAccountId = null

accountInformation = 85512233455

acquiringBank = Dev bank

merchantCategoryCode = 5999

countryCode = KH

merchantName = John Smith

merchantCity = Phnom Penh
```

```
transactionCurrency = 840

transactionAmount = 100.0

billNumber = #12345

mobileNumber = 85512233455

storeLabel = Coffee Shop

terminalLabel = Cashier_1

purposeOfTransaction = Morning coffee

timestamp = 1688006997619

merchantAlternateLanguagePreference = km

merchantNameAlternateLanguage = ចិន ស្ទីន

merchantCityAlternateLanguage = សៀមរាប

crc = 2B5B
```

- **Generate KHQR deeplink**

  Swift

```swift
let deeplinkResponse =
BakongKHQR.generateDeepLink("http://api.example.com/v1/generate_deeplink_by_qr",

qr:"00020101021229180014jonhsmith@nbcq52045999530384054031.05802KH5910Jo
nh Smith6010Phnom Penh6304C297",

            sourceInfo: SourceInfo(appIconURL:"http://cdn.example.com/icons.logo.png",

                    appName: "Example App",

                    appDeepLinkCallBack: "http://app.example.com")

    )
if  deeplinkResponse.status.code == 0 {

    let deeplinkData = deeplinkResponse?.data as? KHQRDeepLinkData

    print(deeplinkData?.shortLink)

} else {

    print(deeplinkResponse.status.message)

}
```

Objective C

```objc
KHQRResponse* deeplinkResponse = [BakongKHQR generateDeepLink:
@"http://api.example.com/v1/generate_deeplink_by_qr"

qr:
@"00020101021229180014jonhsmith@nbcq52045999530384054031.05802KH5910Jo
nh Smith6010Phnom Penh6304C297"

sourceInfo: [[SourceInfo alloc]

                    initWithAppIconURL:@"http://cdn.example.com/icons.logo.png"

                     appName:@"Example App"

                    appDeepLinkCallBack:@"http://app.example.com"]

            ];

if (deeplinkResponse.status.code == 0) {

      KHQRDeepLinkData* deeplinkData = (KHQRDeepLinkData *) deeplinkResponse.data;

      NSLog(@"%@", deeplinkData.shortLink);

} else {

      NSLog(@"%@",deeplinkResponse.status.message);

}
```

Result

```
https://bakongsit.page.link/pBJzebGwUuQfMfhf9
```

*** Note: Generated short links will be different every time method is called
and also depend on which api url being used.

Java

- **Generate KHQR for Individual**

```java
IndividualInfo individualInfo = new IndividualInfo();

individualInfo.setBakongAccountId("john_smith@devb");

individualInfo.setAccountInformation("85512233455");

individualInfo.setAcquiringBank("Dev Bank");

individualInfo.setCurrency(KHQRCurrency.USD);

individualInfo.setAmount(100.0);

individualInfo.setMerchantName("John Smith");

individualInfo.setMerchantCity("PHNOM PENH");
```

```java
individualInfo.setBillNumber("#12345");

individualInfo.setMobileNumber("85512233455");

individualInfo.setStoreLabel("Coffee Shop");

individualInfo.setTerminalLabel("Cashier_1");

 individualInfo.setUpiAccountInformation("1234567812345678ABCDEFGHIJKLMNO");

individualInfo.setPurposeOfTransaction("Buy coffee");

 individualInfo.setMerchantAlternateLanguagePreference("km");

individualInfo.setMerchantNameAlternateLanguage("ចន ស្មីន");

individualInfo.setMerchantCityAlternateLanguage("ភ្នំពេញ");


KHQRResponse<KHQRData> response = BakongKHQR.generateIndividual(individualInfo);

if (response.getKHQRStatus().getCode() == 0) {

    System.out.println("data: " + response.getData().getQr());

    System.out.println("md5: " + response.getData().getMd5());

}
```

Result

data:
00020101021215311234567812345678ABCDEFGHIJKLMNO29460015john_smith@dev
b0111855122334550208Dev Bank520459995303840540310058025KH5910John
Smith6010PHNOM PENH62670106#1234502118551223345503115Coffee
Shop0709Cashier_10810Buy coffee64290002km0108ចន
ស្មីន0207ភ្នំពេញ991700131688023959268630442A9

md5: 0392d8524343ca319a05c54dbdea24e6

- **Generate KHQR for Merchant**

```java
MerchantInfo merchantInfo = new MerchantInfo();

merchantInfo.setBakongAccountId("khqr@devb");

merchantInfo.setMerchantId("123456");

merchantInfo.setAcquiringBank("Dev Bank");

merchantInfo.setCurrency(KHQRCurrency.USD);

merchantInfo.setAmount(100.0);

merchantInfo.setMerchantName("John Smith");
```

```
 merchantInfo.setMerchantCity("PHNOM PENH");

merchantInfo.setBillNumber("#12345");

merchantInfo.setMobileNumber("85512233455");

merchantInfo.setStoreLabel("Coffee Shop");

merchantInfo.setTerminalLabel("Cashier_1");

merchantInfo.setUpiAccountInformation("1234567812345678ABCDEFGHIJKLMNO");

merchantInfo.setPurposeOfTransaction("Buy coffee");

merchantInfo.setMerchantAlternateLanguagePreference("km");

merchantInfo.setMerchantNameAlternateLanguage("ចន ស៊ីន");

merchantInfo.setMerchantCityAlternateLanguage("ភ្នំពេញ");


KHQRResponse<KHQRData> response = BakongKHQR.generateMerchant(merchantInfo);

if (response.getKHQRStatus().getCode() == 0) {

    System.out.println("data: " + response.getData().getQr());

     System.out.println("md5: " + response.getData().getMd5());

}
```

Result

data:
00020101021215311234567812345678ABCDEFGHIJKLMNO30350009khqr@devb0106
1234560208Dev Bank52045999530384054031005802KH5910John Smith6010PHNOM
PENH62670106#123450211855122334550311Coffee Shop0709Cashier_10810Buy
coffee64290002km0108ចន ស៊ីន0207ភ្នំពេញ99170013168802424461863048A5B

md5: e80f432881a2d1b0b271b1cfd755e4b0

● **Verification KHQR**

```
String  qrCode =
"00020101021215311234567812345678ABCDEFGHIJKLMNO30350009khqr@devb010
61234560208Dev Bank52045999530384054031005802KH5910John
Smith6010PHNOM PENH62670106#123450211855122334550311Coffee
Shop0709Cashier_10810Buy coffee64290002km0108ចន
ស៊ីន0207ភ្នំពេញ99170013168802424461863048A5B";

KHQRResponse<CRCValidation> response = BakongKHQR.verify(qrCode);

System.out.println("valid:  " + response.getData().isValid());
```

Result

> valid: true

- **Decode KHQR**

```java
String  qrCode =
"00020101021215311234567812345678ABCDEFGHIJKLMNO30350009khqr@devb010
61234560208Dev Bank520459995303840540310058002KH5910John
Smith6010PHNOM PENH62670106#123450211855122334550311Coffee
Shop0709Cashier_10810Buy coffee64290002km0108ចិន
ស្មីន0207ភ្នំពេញ9917001316880242446186304 8A5B";

KHQRResponse<KHQRDecodeData> response = BakongKHQR.decode(qrCode);

System.out.println(response);
```

Result

```
KHQRResponse{
    status=KHQRStatus{
    code=0,
    errorCode=null,
    message='null' }
 data=KHQRDecodeData{
    payloadFormatIndicator='01',
    pointOfInitiationMethod='12',
    merchantType='30',
    bakongAccountID='khqr@devb',
    merchantId='123456',
    accountInformation='null',
    upiAccountInformation='1234567812345678ABCDEFGHIJKLMNO'
    acquiringBank='Dev Bank',
    merchantCategoryCode='5999',
    countryCode='KH',
    merchantName='John Smith',
    merchantCity='PHNOM PENH',
```

```
    transactionCurrency='840',

    transactionAmount='100',

    billNumber='#12345',

    mobileNumber='85512233455',

    storeLabel='Coffee Shop',

    terminalLabel='Cashier_1',

    purposeOfTransaction='Buy coffee',

    merchantAlternateLanguagePreference='km'

    merchantNameAlternateLanguage='ចន ស្មីន'

    merchantCityAlternateLanguage='ភ្នំពេញ'

    timestamp='1688024244618',

    crc='8A5B'

  }

}
```

● **Generate KHQR deeplink**

```java
String url = "http://api.example.com/v1/generate_deeplink_by_qr";

String qr =

"00020101021229180014jonhsmith@nbcq52045999530384054031.05802KH5910Jonh
 Smith6010Phnom Penh6304C297";

SourceInfo sourceInfo = new SourceInfo();

sourceInfo.setAppName("Example App");

sourceInfo.setAppIconUrl("http://cdn.example.com/icons.logo.png");

sourceInfo.setAppDeepLinkCallback("http://app.example.com");

KHQRResponse<KHQRDeepLinkData> response = BakongKHQR.generateDeepLink(url,
qr, sourceInfo);

if (response.getKHQRStatus().getCode() == 0) {

    System.out.println(response.getData().getShortLink());

}else {

    System.out.println(response.getKHQRStatus().getMessage());

}
```

Result

> https://bakongsit.page.link/8dBnac2EQefMtmsM6

*** **Note:** Generated short links will be different every time method is called and also depend on which api url being used.

C#

- **Generate KHQR For Individual**

```csharp
var response = BakongKHQR.GenerateIndividual(
  new IndividualInfo {
    BakongAccountID = "john_smith@devb",
    Currency = KHQRCurrency.USD,
    AccountInformation = "85512233455",
    AcquiringBank = "Dev Bank",
    Amount = 100,
    MerchantName = "John Smith",
    MerchantCity = "PHNOM PENH",
    BillNumber = "#12345",
    MobileNumber = "85512233455",
    StoreLabel = "Coffee Shop",
    TerminalLabel = "Cashier_1",
    UpiAccountInformation = "1234567812345678ABCDEFGHIJKLMNO",
    PurposeOfTransaction = "Buy coffee",
    MerchantAlternateLanguagePreference = "km",
    MerchantNameAlternateLanguage = "ចន ស្មីន",
    MerchantCityAlternateLanguage = "ភ្នំពេញ"
  });
if (response.Status.Code == 0){
  Console.WriteLine("data: " + response.Data.QR);
  Console.WriteLine("md5: " + response.Data.MD5);
}
```

Result

data:
00020101021215311234567812345678ABCDEFGHIJKLMNO29460015john_smith@devb0111855122334550208Dev Bank52045999530384054031005802KH5910John Smith6010PHNOM PENH62670106#123450211855122334550311Coffee Shop0709Cashier_10810Buy coffee64290002km0108ចិន ស៊ីន0207ភ្នំពេញ991700131688022417673630 41F68

md5: 3655de2d8448ed6fa90cc918469b9a6d

- **Generate KHQR For Merchant**

```
var response = BakongKHQR.GenerateMerchant( new MerchantInfo
  {
      BakongAccountID = "khqr@devb",

      MerchantID = "123456",

      AcquiringBank = "Dev Bank",

      Currency = KHQRCurrency.USD,

      Amount = 100,

      MerchantName = "John Smith",

      MerchantCity = "PHNOM PENH",

      BillNumber = "#12345",

      MobileNumber = "85512233455",

      StoreLabel = "Coffee Shop",

      TerminalLabel = "Cashier_1",

      UpiAccountInformation = "1234567812345678ABCDEFGHIJKLMNO",

      PurposeOfTransaction = "Buy coffee",

      MerchantAlternateLanguagePreference = "km",

      MerchantNameAlternateLanguage = "ចិន ស៊ីន",

      MerchantCityAlternateLanguage = "ភ្នំពេញ"
  });
if (response.Status.Code == 0){

  Console.WriteLine("data: " + response.Data.QR);

  Console.WriteLine("md5: " + response.Data.MD5);

}
```

Result

data:
00020101021215311234567812345678ABCDEFGHIJKLMNO30350009khqr@devb0106
1234560208Dev Bank520459995303840540310005802KH5910John Smith6010PHNOM
PENH62670106#12345021185512233455031101Coffee Shop0709Cashier_10810Buy
coffee64290002km0108ចន ស្មីន0207ភ្នំពេញ99170013168802251535463044A367

md5: cf3696f7dd5bb2c4203df58f6c7ede8f

● **Verification KHQR**

```csharp
var response =
BakongKHQR.Verify("00020101021215311234567812345678ABCDEFGHIJKLMNO3035
0009khqr@devb01061234560208Dev
Bank520459995303840540310005802KH5910John Smith6010PHNOM
PENH62670106#12345021185512233455031101Coffee Shop0709Cashier_10810Buy
coffee64290002km0108ចន ស្មីន0207ភ្នំពេញ99170013168802251535463044A367");

if (response.Status.Code == 0)
{
    Console.WriteLine("valid: " + response.Data.Valid);
}
```

Result:

```
valid: True
```

● **Decode KHQR**

```csharp
var response =
BakongKHQR.Decode("00020101021215311234567812345678ABCDEFGHIJKLMNO303
50009khqr@devb01061234560208Dev
Bank520459995303840540310005802KH5910John Smith6010PHNOM
PENH62670106#12345021185512233455031101Coffee Shop0709Cashier_10810Buy
coffee64290002km0108ចន ស្មីន0207ភ្នំពេញ99170013168802251535463044A367");

var options = new JsonSerializerOptions()
{
    WriteIndented = true
};
```

```
Console.WriteLine(JsonSerializer.Serialize(response, options));
```

Result

```
{
 "Status": {
    "Code": 0,
    "ErrorCode": null,
    "Message": null  },
  "Data": {
   "PayloadFormatIndicator": "01",
   "PointOfInitiationMethod": "12",
   "MerchantType": "30",
   "BakongAccountID": "khqr@devb",
   "MerchantID": "123456",
   "AccountInformation": null,
   "UpiAccountInformation": "1234567812345678ABCDEFGHIJKLMNO",
   "AcquiringBank": "Dev Bank",
   "MerchantCategoryCode": "5999",
   "CountryCode": "KH",
   "MerchantName": "John Smith",
   "MerchantCity": "PHNOM PENH",
   "TransactionCurrency": "840",
   "TransactionAmount": "100",
   "BillNumber": "#12345",
   "MobileNumber": "85512233455",
   "StoreLabel": "Coffee Shop",
   "TerminalLabel": "Cashier_1",
   "PurposeOfTransaction": "Buy coffee",
   "MerchantAlternateLanguagePreference": "km",
   "MerchantNameAlternateLanguage": "ចន ស្មីន",
   "MerchantCityAlternateLanguage": "ភ្នំពេញ",
   "Timestamp": "1688022515354",
   "CRC": "A367"
  }
```

- **Generate KHQR deeplink**

```
var url = "http://api.example.com/v1/generate_deeplink_by_qr";
var qr =
"00020101021229180014jonhsmith@nbcq52045999530384054031.05802KH5910Jonh
Smith6010Phnom Penh6304C297";
var sourceInfo = new SourceInfo {
    AppName = "Example App",
    AppIconUrl = "http://cdn.example.com/icons.logo.png",
    AppDeepLinkCallback = "http://app.example.com"
};
var response = BakongKHQR.GenerateDeepLink(url, qr, sourceInfo);
if (response.Status.Code == 0) {
    Console.WriteLine(response.Data.ShortLink);
} else {
    Console.WriteLine(response.Status.Message);
}
```

Result

```
https://bakongsit.page.link/DqKPTPQ521gXd4g8A
```

*** Note: Generated short links will be different every time method is called and also depend on which api url being used.

Javascript

- **Generate Individual KHQR**

```javascript
const {
 BakongKHQR,
   khqrData,
   IndividualInfo,
   MerchantInfo,
} = require("bakong-khqr");

const optionalData = {
 currency: khqrData.currency.usd,
 amount: 100.5,
 mobileNumber: "85512233455",
 storeLabel: "Coffee Shop",
 terminalLabel: "Cashier_1",
 purposeOfTransaction: "oversea",
 languagePreference: "km",
 merchantNameAlternateLanguage: "ចន ស្ម៉ីន",
 merchantCityAlternateLanguage: "សៀមរាប",
 upiMerchantAccount: "0001034400010344ABCDEFGHJIKLMNO"
};
const individualInfo = new IndividualInfo(
   "jonh_smith@devb",
   "Jonh Smith",
   "PHNOM PENH",
   optionalData);
const KHQR = new BakongKHQR();
const individual = KHQR.generateIndividual(individualInfo);
console.log("qr: " + individual.data.qr);
console.log("md5: " + individual.data.md5);
```

Result

qr:
'00020101021215310001034400010344ABCDEFGHJIKLMNO29190015jonh_smith@devb520459995303840540610 0.505802KH5910Jonh Smith6010PHNOM PENH6254021185512233455 0311Coffee Shop0709Cashier_10807oversea64280002km0108ចិន ស្មៀន0206សៀមរាប99170013168853836199963044964',

md5: 'cdbc245955a9ff043a7b304b3b742894'

- **Generate Merchant KHQR**

```
const {
    BakongKHQR,
    khqrData,
    IndividualInfo,
    MerchantInfo,
} = require("bakong-khqr");
const optionalData = {
 currency: khqrData.currency.usd,
 amount: 100.5,
 mobileNumber: "85512233455",
 storeLabel: "Coffee Shop",
 terminalLabel: "Cashier_1",
 purposeOfTransaction: "oversea",
 languagePreference: "km",
 merchantNameAlternateLanguage: "ចិន ស្មៀន",
 merchantCityAlternateLanguage: "សៀមរាប",
 upiMerchantAccount: "0001034400010344ABCDEFGHJIKLMNO"
};
const merchantInfo = new MerchantInfo(
    "khqr@devb",
    "Jonh Smith",
    "PHNOM PENH",
    "123456",
    "Dev Bank",
    optionalData
```

```
);
const KHQR = new BakongKHQR();
const merchant = KHQR.generateMerchant(merchantInfo);
console.log("qr: " + merchant.data.qr);
console.log("md5: " + merchant.data.md5);
```

Result

```
qr:
'00020101021215310001034400010344ABCDEFGHJIKLMNO30350009khqr@devb010
61234560208Dev Bank520459995303840540610.505802KH5910Jonh
Smith6010PHNOM PENH625402118551223345503110311Coffee
Shop0709Cashier_10807oversea64280002km0108ចិន
ស្មីន0206សៀមរាប991700131688538361998 6304469E',

md5: '0d1806c4e56f066bfaa862c26af21bec'
```

- **Verification KHQR**

```
let khqrString = "00020101021230350009khqr@devb01061234560208Dev
Bank520459995303840540610.505802KH5910Jonh Smith6010PHNOM
PENH62430211855122334550311Coffee
Shop0709Cashier_199170013168448026228463040009F";
const isKHQR = BakongKHQR.verify(khqrString);
console.log("valid:" + isKHQR.isValid)
```

Result

```
valid:true
```

- **Decode KHQR**

```
let khqrString =
"000201010212153111234567812345678ABCDEFGHIJKLMNO30350009khqr@devb010
61234560208Dev Bank520459995303840540310055802KH5910John
Smith6010PHNOM PENH62670106#12345021185512233455031105311Coffee
Shop0709Cashier_10810Buy coffee64290002km0108ចិន
ស្មីន0207ភ្នំពេញ9917001316880225153546304A367";
```

```
const decodeValue = BakongKHQR.decode(khqrString)

console.log(decodeValue)
```

Result

```
{
  status: { code: 0, errorCode: null, message: null },
  data: {
    merchantType: '30',
    bakongAccountID: 'khqr@devb',
    accountInformation: null,
    merchantID: '123456',
    acquiringBank: 'Dev Bank',
    billNumber: '#12345',
    mobileNumber: '85512233455',
    storeLabel: 'Coffee Shop',
    terminalLabel: 'Cashier_1',
    purposeOfTransaction: 'Buy coffee',
    languagePreference: 'km',
    merchantNameAlternateLanguage: 'ចិន ស៊ីន',
    merchantCityAlternateLanguage: 'ភ្នំពេញ',
    payloadFormatIndicator: '01',
    pointofInitiationMethod: '12',
    unionPayMerchant: '1234567812345678ABCDEFGHIJKLMNO',
    merchantCategoryCode: '5999',
    transactionCurrency: '840',
    transactionAmount: '100',
    countryCode: 'KH',
    merchantName: 'John Smith',
    merchantCity: 'PHNOM PENH',
    timestamp: '00131688022515354',
    crc: 'A367'
  }
}
```
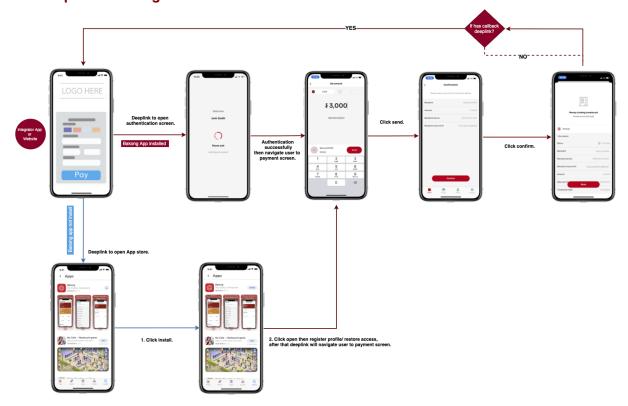
● **Generate KHQR deeplink**

```
const khqrString =
"00020101021229190015john_smith@devb5204599953038405405100.05802KH5910J
ohn Smith6010Phnom Penh6304BF30";

const sourceInfo = new SourceInfo(

    "http://cdn.example.com/icons.logo.png",

    "Example App",

    "http://app.example.com");

const deeplink = khqr.generateDeepLink(

    "http://api.example.com/v1/generate_deeplink_by_qr",

    khqrString,

    sourceInfo

);

deeplink.then((data) => {

  if (data.status.code == 0) {

    console.log(data.data.shortLink);

  } else {

    console.log(data.status.message);

  }

});
```

Result

```
https://bakongsit.page.link/16ahB249Xu9TQc676
```

*** Note: Generated short links will be different every time method is called
and also depend on which api url being used.

Deeplink Diagram

Deeplink Rounting

YES

If has callback deeplink?

'NO'

Integrator App or Website

LOGO HERE

Pay

Deeplink to open authentication screen.

Bakong App installed

Bakong app not install

Deeplink to open App store.

Welcome

Jonh Smith

Please wait

Set amount

₿ 3,000

Add description

Send

Authentication successfully then navigate user to payment screen.

Click send.

Confirmation

Confirm

Click confirm.

Money is being transferred

Done

< Apps

Bakong

My Cafe — Restaurant game

1. Click install.

< Apps

Bakong

My Cafe — Restaurant game

2. Click open then register profile/ restore access, after that deeplink will navigate user to payment screen.

Standard Response

Response Format

```
{
  "status": {
      "code": 0,
      "errorCode": null,
      "message": 'This is message'
  },
  "data": {
      //response based on each function
          qr: "00020101021229460015john_smith@devb0111855122334550208Dev
Bank520459995303840540310058020KH5910John Smith6010Phnom
Penh62500106#1234502118551223345503110Coffee
Shop0706012345991700131631170385549630480C"
  }
```

Response Key and Code with definition

Key	Type	Value	Message
code	int		
		0	Success
		1	Failed
errorCode	int		
		1	Bakong Account ID cannot be null or empty
		2	Merchant name cannot be null or empty
		3	Bakong Account ID is invalid
		4	Amount is invalid
		5	Merchant type cannot be null or empty
		6	Bakong Account ID Length is invalid
		7	Merchant Name Length is invalid
		8	KHQR provided is invalid
		9	Currency type cannot be null or empty

		10	Bill Number Length is invalid
		11	Store Label Length is invalid
		12	Terminal Label Length is invalid
		13	Cannot reach Bakong Open API service. Please check internet connection
		14	Source Info for Deep Link is invalid
		15	Internal Server Error
		16	Payload Format Indicator Length is invalid
		17	Point of Initiation Length is invalid
		18	Merchant Category Length is invalid
		19	Transaction Currency Length is invalid
		20	Country Code Length is invalid
		21	Merchant City Length is invalid
		22	CRC Length is invalid
		23	Payload Format Indicator cannot be null or empty
		24	CRC cannot be null or empty
		25	Merchant Category cannot be null or empty
		26	Country Code cannot be null or empty
		27	Merchant City cannot be null or empty
		28	Unsupported currency
		29	Deep Link URL is not valid
		30	Merchant ID cannot be null or empty
		31	Acquiring Bank cannot be null or empty
		32	Merchant ID Length is invalid
		33	Acquiring Bank Length is invalid
		34	Mobile Number Length is invalid
		35	Tag not in order
		36	Account information length is invalid

		37	Merchant Alternate Language Preference cannot be null or empty
		38	Merchant Alternate Language Preference Length is invalid
		39	Merchant Name Alternate Language cannot be null or empty
		40	Merchant Name Alternate Language Length is invalid
		41	Merchant City Alternate Language Length is invalid
		42	Purpose of Transaction Length is invalid
		43	Upi Account Information Length is invalid
message	String		The message describes the result.
data	Any Object		The response will be different based on function
qr	String		emv qr code provided by KHQR library.
md5	String		hash value of bakong KHQR to verify transaction status.

FAQ

1. If my bank wants to know more or inquire about KHQR, who should I talk to?

KHQR team of Association of Bank in Cambodia

2. If my bank wants to know more or inquire about How To Implement KHQR By Using Bakong As a Payment Switch, who should I talk to?

Bakong team of National Bank of Cambodia